The following security alert was issued by the Information Security Division of the Mississippi Department of ITS and is intended for State government entities. The information may or may not be applicable to the general public and accordingly, the State does not warrant its use for any specific purposes.

TLP: WHITE
Traffic Light Protocol (TLP): WHITE information may be distributed without restriction, subject to copyright controls.
http://www.us-cert.gov/tlp/

DATE(S) ISSUED:
07/22/2016

SUBJECT:
Multiple Vulnerabilities in PHP Could Allow For Arbitrary Code Execution

OVERVIEW:
Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow an attacker to execute arbitrary code, with failed exploit attempts potentially leading to denial of service conditions. PHP is a programming language originally designed for use in web-based applications with HTML content. PHP supports a wide variety of platforms and is used by numerous web-based software applications. Successfully exploiting these issues could allow remote attackers to execute arbitrary code in the context of the affected application. Failed exploitation could result in a denial-of-service condition.

THREAT INTELLIGENCE:
There are currently no reports of these vulnerabilities being exploited in the wild. There is known proof-of-concept code for these vulnerabilities.

SYSTEM AFFECTED:

- PHP 5.5 prior to 5.5.38
- PHP 5.6 prior to 5.6.24
- PHP 7 prior to 7.0.9

RISK:
Government:

- Large and medium government entities: **High**
- Small government entities: **High**

Businesses:

- Large and medium business entities: **High**
- Small business entities: **High**

Home users: Low

TECHNICAL SUMMARY:
PHP has released updates that address multiple vulnerabilities, the most severe of which could allow for arbitrary code execution. These vulnerabilities include:

Prior to 5.5.38

- Bug #72613 - Inadequate error handling in bzread().
- Bug #70480 - php_url_parse_ex() buffer overflow read.
- Bug #72513 - Stack-based buffer overflow vulnerability in virtual_file_ex.

- Bug #72562 - Use After Free in unserialize() with Unexpected Session Deserialization.
- Bug #72573 - HTTP_PROXY is improperly trusted by some PHP libraries and applications.
- Bug #72603 - Out of bound read in exif_process_IFD_in_MAKERNOTE.
- Bug #72618 - NULL Pointer Dereference in exif_process_user_comment.
- Bug #72512 - gdImageTrueColorToPaletteBody allows arbitrary write/read access.
- Bug #72519 - imagegif/output out-of-bounds access.
- Bug #72558 - Integer overflow error within _gdContributionsAlloc().
- Bug #72533 - locale_accept_from_http out-of-bounds access.
- Bug #69975 - PHP segfaults when accessing nvarchar(max) defined columns.
- Bug #72479 - Use After Free Vulnerability in SNMP with GC and unserialize().
- Bug #72606 - heap-buffer-overflow (write) simplestring_addn simplestring.c.
- Bug #72520 - Stack-based buffer overflow vulnerability in php_stream_zip_opener.

Prior to 5.6.24

- Bug #71936 - Segmentation fault destroying HTTP_RAW_POST_DATA.
- Bug #72496 - Cannot declare public method with signature incompatible with parent private method.
- Bug #72138 - Integer Overflow in Length of String-typed ZVAL.
- Bug #72513 - Stack-based buffer overflow vulnerability in virtual_file_ex.
- Bug #72562 - Use After Free in unserialize() with Unexpected Session Deserialization.
- Bug #72573 - HTTP_PROXY is improperly trusted by some PHP libraries and applications.
- Bug #72447 - Type Confusion in php_bz2_filter_create(). (gogil at stealien dot com.
- Bug #72613 - Inadequate error handling in bzread().
- Bug #50845 - exif_read_data() returns corrupted exif headers.
- Bug #72603 - Out of bound read in exif_process_IFD_in_MAKERNOTE.
- Bug #72618 - NULL Pointer Dereference in exif_process_user_comment.
- Bug #43475 - Thick styled lines have scrambled patterns.
- Bug #53640 - XBM images require width to be multiple of 8.
- Bug #64641 - imagefilledpolygon doesn't draw horizontal line.
- Bug #72512 - gdImageTrueColorToPaletteBody allows arbitrary write/read access.
- Bug #72519 - imagegif/output out-of-bounds access.
- Bug #72558 - Integer overflow error within _gdContributionsAlloc().
- Bug #72533 - locale_accept_from_http out-of-bounds access.
- Bug #69975 - PHP segfaults when accessing nvarchar(max) defined columns.
- Bug #71915 - openssl_random_pseudo_bytes is not fork-safe.
- Bug #72336 - openssl_pkey_new does not fail for invalid DSA params.
- Bug #72479 - Use After Free Vulnerability in SNMP with GC and unserialize().
- Bug #55701 - GlobIterator throws LogicException.
- Bug #70628 - Clearing bindings on an SQLite3 statement doesn't work.
- Bug #72439 - Stream socket with remote address leads to a segmentation fault.
- Bug #72606 - heap-buffer-overflow (write) simplestring_addn simplestring.c.
- Bug #72520 - Stack-based buffer overflow vulnerability in php_stream_zip_opener.

Prior to 7.0.9

- Bug #72508 - strange references after recursive function call and "switch" statement.
- Bug #72513 - Stack-based buffer overflow vulnerability in virtual_file_ex.
- Bug #72573 - HTTP_PROXY is improperly trusted by some PHP libraries and applications.
- Bug #72613 - Inadequate error handling in bzread().

- Bug #72484 - SCRIPT_FILENAME shows wrong path if the user specify router.php.
- Bug #72498 - variant_date_from_timestamp null dereference.
- Bug #72541 - size_t overflow lead to heap corruption.
- Bug #72603 - Out of bound read in exif_process_IFD_in_MAKERNOTE.
- Bug #72618 - NULL Pointer Dereference in exif_process_user_comment.
- Bug #43475 - Thick styled lines have scrambled patterns.
- Bug #53640 - XBM images require width to be multiple of 8.
- Bug #64641 - imagefilledpolygon doesn't draw horizontal line.
- Bug #72512 - gdImageTrueColorToPaletteBody allows arbitrary write/read access.
- Bug #72519 - imagegif/output out-of-bounds access.
- Bug #72558 - Integer overflow error within _gdContributionsAlloc().
- Bug #72482 - Ilegal write/read access caused by gdImageAALine overflow.
- Bug #72494 - imagecropauto out-of-bounds access.
- Bug #72533 - locale_accept_from_http out-of-bounds access.
- Bug #72405 - mb_ereg_replace - mbc_to_code (oniguruma) - oob read access.
- Bug #72399 - Use-After-Free in MBString (search_re).
- Bug #72551 - bug #72552 (Incorrect casting from size_t to int lead to heap overflow in mdecrypt_generic.
- Bug #72570 - Segmentation fault when binding parameters on a query without placeholders.
- Bug #72476 - Memleak in jit_stack.
- Bug #72463 - mail fails with invalid argument.
- Bug #72538 - readline_redisplay crashes php.
- Bug #72505 - readfile() mangles files larger than 2G.
- Bug #72306 - Heap overflow through proc_open and $env parameter.
- Bug #72531 - ps_files_cleanup_dir Buffer overflow.
- Bug #72562 - Use After Free in unserialize() with Unexpected Session Deserialization.
- Bug #72479 - Use After Free Vulnerability in SNMP with GC and unserialize().
- Bug #72439 - Stream socket with remote address leads to a segmentation fault.
- Bug #72606 - heap-buffer-overflow(write) simplestring_addn simplestring.c.
- Bug #72520 - Stack-based buffer overflow vulnerability in php_stream_zip_opener.

Successfully exploiting these issues could allow remote attackers to execute arbitrary code in the context of the affected application. Failed exploitation could result in a denial-of-service condition.

RECOMMENDATIONS:
The following actions should be taken:

- Upgrade to the latest version of PHP immediately, after appropriate testing.
- Apply the principle of Least Privilege to all systems and services.
- Verify no unauthorized system modifications have occurred on system before applying patch.
- Remind users not to visit websites or follow links provided by unknown or untrusted sources.
- Do not open email attachments from unknown or untrusted sources.
- Limit user account privileges to only those required.

References:
NOTE: Visiting these links may trigger an IDS signature match for a Possible Encrypted Webshell Download. This is a false positive alert that is matching content on the pages below.

PHP:
http://php.net/ChangeLog-5.php#5.6.24

http://php.net/ChangeLog-5.php#5.5.38
http://php.net/ChangeLog-7.php